

# **Instruction Set of 8085 Microprocessor**

# Instruction Set of 8085 Microprocessor

- An instruction is a binary pattern designed inside a microprocessor to perform a specific function.
- The entire group of instructions that a microprocessor supports is called **Instruction Set**.
- 8085 has **246** instructions.
- Each instruction is represented by an **8-bit** binary value.
- These 8-bits of binary value is called **Op-Code** or **Instruction Byte**.

An instruction is a command given to the microprocessor to perform a specific operation on data.

Basically there are **five groups** of instructions –

- **Data transfer instructions** – these instructions are used to transfer the data from one register to another register or memory to register etc.
- **Arithmetic instructions** – these instructions are going to perform arithmetic operations like addition, subtraction etc.
- **Logical instructions** – these instructions are going to perform logical operations like AND, OR, XOR etc.
- **Branching instructions** – these instructions are going to design inside the program controller where we need to jump the program control. Example: JUMP, CALL, RETURN etc.
- **I/O and Machine control instructions** – these instruction contains I/O and machine related contents.

## Data Transfer Instructions:

These are used for data transfer from one place to another place.

Instruction	Operation
MOV R1, R2	R1 → R2
MOV M, R (M points to HL register pair)	M → R
MOV R, M	R → M
MVI R, FFH	R ← FFH
MVI R, 8-bit data	R ← 8-bit data
LXI SP, 8-bit data	SP ← 8-bit data

# Arithmetic Instructions

<b>Instruction</b>	<b>Operation</b>	<b>Instruction</b>	<b>Operation</b>	<b>Instruction</b>	<b>Operation</b>
ADD R	$A \leftarrow A+R$	ADC R	$A \leftarrow A+R+CY$	INR R	$R \leftarrow R+1$
ADD M	$A \leftarrow A+M$	ADC M	$A \leftarrow A+M+CY$	INR M	$M \leftarrow M+1$
ADI 8-bit data	$A \leftarrow A+8\text{-bit data}$	ADC 8-bit data	$A \leftarrow A+8\text{-bit data}+CY$	DCR R	$R \leftarrow R-1$
SUB R	$A \leftarrow A-R$	SBB R	$A \leftarrow A-R-CY$	DCR M	$M \leftarrow M-1$
SUB M	$A \leftarrow A-M$	SBB M	$A \leftarrow A-M-CY$	INX rp	$rp \leftarrow rp+1$
SUI 8-bit data	$A \leftarrow A-8\text{ bit data}$	SBI 8-bit data	$A \leftarrow A-8\text{-bit data}-CY$	DCX rp	$rp \leftarrow rp-1$

# Logical Instructions

Instruction	Operation	Instruction	Operation	Instruction	Operation
ANA R	$A \leftarrow A \wedge R$	XRA R	$A \leftarrow A \oplus R$	RLC	Rotate Left without carry
ANA M	$A \leftarrow A \wedge M$	XRA M	$A \leftarrow A \oplus M$	RRC	Rotate Right without carry
ANI 8-bit data	$A \leftarrow A \wedge 8\text{-bit data}$	XRA 8-bit data	$A \leftarrow A \oplus 8\text{-bit data}$	RAL	Rotate Left with carry
ORA R	$A \leftarrow A \vee R$	CMP R	$A \leftarrow A - R$	RAR	Rotate Left with carry
ORA M	$A \leftarrow A \vee M$	CMP M	$A \leftarrow A - M$	CMA	Complement Accumulator
ORI 8-bit data	$A \leftarrow A \vee 8\text{-bit data}$	CPI 8-bit data	$A \leftarrow A - 8\text{-bit data}$	CMC	Complement Carry

## Branching Instructions

We are having three types of branching instructions.

- Jump (Unconditional and Conditional) – Example: JUMP 16 bit address
- Call (Unconditional and Conditional) – Example: CALL 16 bit address (PUSH)
- Return (Unconditional and Conditional) – Example: RET (POP)

Condition depends on flag registers.

- NZ (Z=0), Z (Z=1) C (C=1) NC (C=0) PE (P=1) PO (P=0) P(Plus) (S=0) M (Minus) (S=1)
- In every JUMP instruction Program counter (PC) is going to catch the 16 bit address
- Ex: JNZ, JZ, JC, JNC, JPE, JPO, JP, JM
- In every CALL instruction Stack Pointer (SP) is decremented by 2
- Ex: CNZ, CZ, CC, CNC, CPE, CPO, CP, CM
- In every RET instruction Stack Pointer (SP) is incremented by 2
- Ex: RNZ, RZ, RC, RNC, RPE, RPO, RP, RM

## Stack, I/O and Machine Control Instructions

This is the last category of instruction set in 8085 microprocessor which is having totally 10 instructions into 3 categories.

**Stack:** i. **PUSH** – Push two bytes of data into stack.

ii. **POP** – Pop two bytes of data from stack.

iii. **HThL** – Exchange top of stack with HL.

iv. **SPHL** – Move contents of HL to stack.

**I/O:** v. **IN** – Initiate input operation.      Example: IN 8 bit data A 8 bit data



vi. **OUT** – Initiate output operation.      Example: OUT 8 bit data A 8 bit data

**Machine Control:** vii. **EI** – Enable interrupt      All interrupts are enabled

viii. **DI** – Disable interrupt      RST 7.5, 6.5, 5.5 and INTR will be disabled

ix. **HLT** – Microprocessor is halted

x. **NoP** – No operation